UNIVERSITY OF MANNHEIM
School of Business Informatics and Mathematics

Chair of Practical Computer Science IV:
Dependable Systems Engineering

Master's Thesis

Vulnerabilities in Privacy-Preserving Record Linkage: The Threat of Dataset Extension Attacks

as part of the degree program Master of Science Business Informatics submitted by

Marcel Mildenberger
Matriculation number 1979905

on March 6, 2025.

Supervisor: Prof. Dr. Frederik Armknecht
PhD Student Jochen Schäfer

Abstract

The abstract should serve as an independent piece of information on your Thesis conveying a concise description of the main aspects and most important results. It should not be excessively long.

Write the abstract.

Contents

List of Figures

List of Tables

List of Algorithms

List of Code Snippets

Acronyms

AI Artifical Intelligence

ANN Artificial Neural Network

BF Bloom Filter

DEA Dataset Extension Attack

ELD Encoded Linkage Data

GMA Graph Matching Attack

ML Machine Learning

PII Personally Identifiable Information

PPRL Privacy-Preserving Record Linkage

PRNG Pseudo-Random Number Generator

SMC Secure Multi-Party Computation

TMH Tabulation MinHash Encoding

TSH Two-Step Hash Encoding

1. Introduction

Linking data and records is an important component of research, software development and software projects. The primary reason for integrating data from different sources is to gain richer, more comprehensive insights about the same entity. Initially, deterministic record linkage, which relies on exact matches between predefined identifiers such as unique IDs, was the main method used in early linkage techniques. However, deterministic approaches often fail in real-world scenarios where data may suffer from inconsistent formatting, typographical errors or missing values, making exact matches impossible [HSW07].

The introduction of a probabilistic framework for record linkage by Fellegi et al. in 1969 [FS69] marked a significant advance in overcoming these limitations. Their work, "A Theory for Record Linkage" [FS69], proposed a statistical method for linking records across datasets by calculating the probability that two records refer to the same entity, even when there are inconsistencies in the data. Their approach evaluates common attributes and assigns weights based on the likelihood of a match as opposed to a random similarity. By accounting for discrepancies in real-world data, the so-called Fellegi-Sunter model has become a fundamental methodology for data linkage, particularly in heterogeneous and distributed data environments where traditional deterministic methods fall short.

Such a probabilistic approach to record linkage is important in sectors such as healthcare and social sciences, where data is often distributed across multiple institutions or sources and lacks unique identifiers. In these fields, the ability to integrate datasets is essential for gaining insights and improving outcomes. In the United States, for example, the healthcare system is highly fragmented, consisting of numerous independent entities such as hospitals, clinics, insurance companies, public health agencies, and research institutions. Each of these organisations collects and stores patient data independently, often using different systems and, more importantly, different formats. This fragmentation creates significant challenges when trying to track patient outcomes, monitor disease outbreaks, or evaluate the effectiveness of treatments across populations. Effective data linkage can bridge these gaps by linking records that refer to the same individual across multiple datasets. This integration is critical for tasks such as epidemiological research, public health surveillance, personalised medicine and healthcare quality improvement [PSZ+24; VSCR17].

For example, during the COVID-19 pandemic, the inability to efficiently link data between testing centres, hospitals and vaccination sites limited timely tracking of infection rates and vaccination outcomes. Had more robust data linkage mechanisms been in place, public health officials could have responded more effectively to outbreaks and targeted interventions to specific populations. Data linkage thus plays a critical role in transforming fragmented data landscapes into unified and actionable insights, leading to more informed decision-making. In response to the COVID-19 pandemic, organisations such as the Centers for Disease Control and Prevention and the Food and Drug Administration have launched projects to address these challenges and further develop linkage techniques [PSZ+24].

In scenarios such as the COVID-19 pandemic, data integration efforts often involve linking records on natural persons from multiple sources. For example, integrating data from differ-

ent healthcare providers, laboratories and public health agencies typically requires the use of pseudo-identifiers derived from Personally Identifiable Information (PII), such as names, dates of birth or other sensitive information. However, reliance on PII for linkage raises significant privacy concerns, as improper handling of such data can lead to re-identification of individuals, with potentially serious consequences such as data breaches, identity theft or unauthorised access to personal health information [PSZ+24; SBR09].

The increasing digitisation of personal data has already led to large-scale data breaches, demonstrating the risks of improperly secured data. Notable incidents such as the Cambridge Analytica scandal, in which personal data was misused for political profiling, highlight the ethical and regulatory challenges of data integration [IH18]. Similarly, healthcare data leaks have raised concerns about the implications of unauthorised access to medical histories, genetic data and insurance records. Leaks of personal health information can have serious consequences, including blackmail, discrimination, and fraud, which can cause significant personal harm. For example, individuals whose medical histories are exposed may face discrimination in employment or insurance, while others may become targets of scams that exploit their health conditions. The potential for such abuse underlines the critical importance of robust data protection measures by working with PII [Smi16].

To address these privacy risks, various techniques have been developed to protect PII during the linking process, primarily by encrypting the data prior to linking. However, the use of encrypted PII as pseudo-identifiers presents additional challenges. The key question is how to efficiently encrypt sensitive information while maintaining the ability to accurately match records [SBR09].

Therefore, Privacy-Preserving Record Linkage (PPRL) techniques are designed to facilitate data integration without exposing sensitive information, ensuring that datasets can be securely linked across different entities. To enable linkage while preserving privacy, similarity preserving encryption is applied to the PII. Without such similarity preserving encryption, matches between encrypted entities in different databases would not be possible [SBR09; VSCR17].

Over time, three main privacy-preserving encryption schemes have emerged as enablers for PPRL [SAH24; VCRS20].

Bloom Filter (BF) encoding is the most widely used technique in PPRL and is often considered the reference standard [SAH24]. Originally introduced by Burton Bloom in 1970 as a probabilistic data structure for efficient set membership testing [Blo70], BFs were later adapted for PPRL due to their simplicity and efficiency in both storing and computing set similarities. Their compact representation and probabilistic nature make them ideal for scalable PPRL systems, especially in environments dealing with large datasets [SBR09]. The seminal work of Schnell et al. demonstrated the use of BFs in PPRL, particularly in healthcare, highlighting their ability to perform secure record matching without exposing sensitive identifiers [SBR09]. However, BFs are not without limitations. Their vulnerability to graph-based attacks and pattern exploitation has driven research into improving their security. Techniques such as diffusion have been proposed to obscure recognisable patterns and increase security [AHS23; SAH24]. For example, Armknecht et al. [AHS23] explored methods to strengthen the security of BF by adding a linear diffusion layer to the BF-based PPRL approach, which complicates pattern mining attacks.

To address some of these weaknesses of BFs, Tabulation MinHash Encoding (TMH) has been introduced as a more secure alternative. MinHash, first developed by Broder in 1997 for estimating set similarities in large document collections [Bro97], has been adapted using tabulation-based hashing [Smi17]. Although less widely used than BFs, TMH offers distinct

advantages, including stronger security guarantees against re-identification attacks. However, these benefits come at the cost of increased computational complexity and memory usage, which may limit its applicability in resource-constrained environments [Smi17].

A further development in encryption techniques is the introduction of Two-Step Hash Encoding (TSH), which aims to combine the strengths of both BFs and TMH while mitigating their respective weaknesses. As detailed by [RCS20], TSH employs a two-stage process: data is first encrypted using multiple BFs, followed by an additional hashing layer that transforms the encrypted data into a set of integers suitable for similarity comparison. This layered approach enhances privacy by adding an extra layer of obfuscation, making it more resistant to attack, while maintaining efficient similarity computations [RCS20; VCRS20].

In practice, BF-based PPRL has become the dominant standard and is widely used in areas such as crime detection, fraud prevention and national security due to its balance of efficiency and ease of implementation. However, BF-based PPRL systems are not without limitations and vulnerabilities. Previous research has shown that there are several attacks targeting PPRL systems, with a focus on exploiting the weaknesses inherent in BF encodings. These attacks specifically target weaknesses in BF constructions, such as the weaknesses introduced by double hashing, structural flaws in filter design, and susceptibility to common pattern-mining techniques. Notably, no specific attacks have been developed for TMH or TSH encodings, suggesting that research has focused primarily on the more widely used BF scheme. [VCRS20]

However, a more recent and practical attack has emerged that exploits vulnerabilities common to all PPRL encryption schemes. The Graph Matching Attack (GMA) uses publicly available data, such as telephone directories, to re-identify encrypted individuals based on overlapping records between plaintext and encrypted databases [SAH24; VCRS20]. Unlike previous attacks that focus solely on the encryption scheme of BFs, the GMA works independently of the encryption scheme chosen. It therefore exploits the graph structure of encoded datasets to re-identify records. Given two datasets - a plaintext reference dataset and an encrypted dataset - an attacker can construct similarity graphs where nodes represent individuals and edges represent similarity scores. By solving a graph isomorphism problem, attackers can infer one-to-one mappings between encrypted and plaintext records, effectively breaking the privacy guarantees of PPRL. The effectiveness of GMAs depends on the overlap between the two sets of data; the greater the overlap, the higher the probability of successful re-identification. While GMAs can successfully re-identify individuals present in both the plaintext and encrypted datasets, their effectiveness is limited to the overlapping subset of the two databases [SAH24; VCRS20].

This work aims to go beyond traditional GMAs by re-identifying not only individuals present in the overlapping datasets, but as many individuals as possible from the encrypted PPRL data. To achieve this, the newly introduced Dataset Extension Attack (DEA) builds on the foundations laid by GMAs. The DEA uses an Artificial Neural Network (ANN) trained on the subset of previously re-identified individuals to predict and decode the remaining encrypted records. In doing so, the DEA significantly expands the scope and effectiveness of the attack, enabling broader de-anonymisation of PPRL datasets beyond the limitations of existing graph-based methods.

1.1. Motivation

The increasing use of PPRL in highly sensitive areas such as healthcare, finance and national security requires research to validate existing techniques and ensure robust privacy [SBR09]. As data-driven applications continue to evolve, the complexity and volume of data being collected and linked across multiple sources is growing rapidly. While PPRL systems are designed to facilitate secure data integration without compromising privacy, evolving cybersecurity threats and attack techniques highlight the urgent need to reassess the resilience of these systems [VSCR17].

Privacy has always been a critical concern in data management, but its importance has been intensified in the era of Artifical Intelligence (AI) and Machine Learning (ML). These technologies increasingly rely on large data sets, often containing sensitive PII such as medical records, financial transactions or behavioural data for training. If compromised, the exposure of such data can lead to serious privacy violations, including identity theft, financial fraud and discrimination. The rise of data brokerage, where personal information is collected, aggregated and sold - often without explicit user consent - further exacerbates privacy concerns. This commoditisation of personal data has made PII an attractive target for malicious actors, increasing the risk of unauthorised data linking and re-identification attacks. As AI models become more advanced, the demand for rich, high-quality data continues to grow, making privacy an increasingly pressing issue [KM24; MK19].

In this context, the vulnerability of PPRL systems to emerging attack methods is of particular concern. While PPRL techniques such as BF are designed to hide sensitive identifiers during the data linkage process, recent research has shown that these systems are vulnerable to GMAs. GMAs exploit the similarity preserving properties of common encryption schemes to re-identify individuals by comparing patterns in encrypted records with those in publicly available plaintext records. This approach undermines the fundamental goal of PPRL: to protect sensitive data during the record linkage process. Although current GMAs are limited to re-identifying individuals present in both the encrypted and plaintext datasets, even partial data exposure in highly sensitive areas can have serious consequences [SAH24; VCRS20].

The introduction of DEAs poses an even greater threat to the integrity of PPRL systems. Unlike GMAs, DEAs aim to extend the scope of re-identification to as many individuals as possible within the encrypted database. Using ANNs trained on previously decoded data from GMAs, DEAs can predict and decode additional records, potentially leading to the complete deanonymisation of entire encrypted datasets. This represents a paradigm shift, as it challenges the viability of widely used PPRL techniques, such as BF-based encryption, which have been considered secure.

The primary motivation for this research is to proactively investigate and demonstrate the consequences of such advanced attacks in order to prevent their realisation in real-world scenarios. By exposing the potential vulnerabilities of PPRL systems, this work aims to demonstrate how attackers could exploit decrypted data to compromise privacy on a large scale. A successful implementation of the DEA will provide empirical evidence that state-of-the-art methods are insufficiently secure, highlighting the urgent need for more robust privacy-preserving techniques.

Furthermore, there is a notable gap in current research regarding the extension of attack capabilities beyond the intersection of datasets. While significant efforts have been made to address the vulnerabilities exposed by GMAs, there is a lack of comprehensive studies exploring how ML can be used to generalise these attacks and compromise entire databases. This research

aims to fill this gap by developing and evaluating the DEA, thereby contributing to a broader understanding of PPRL vulnerabilities.

By addressing this gap, this thesis aims to contribute to the body of knowledge on PPRL vulnerabilities and serve as a foundation for future research aimed at strengthening these systems. The knowledge gained from this study will not only enable the development of more secure PPRL techniques, but will also influence best practices in privacy and security.

1.2. Related Work

The study by Vidanage et al. [VCRS20] represents a significant advance in the field of PPRL through the introduction of a new attack method known as GMA. Their work begins with a comprehensive overview of PPRL systems and the similarity preserving encryption techniques commonly used, such as BFs. The GMA exploits weaknesses in these encoding schemes by exploiting their ability to preserve partial similarity information even after encryption. By constructing similarity graphs from both encrypted and plaintext datasets, the GMA solves a graph isomorphism problem to align nodes and successfully re-identify individuals in the encrypted dataset using publicly available sources such as telephone directories. This method demonstrates the universal applicability of GMAs across different PPRL schemes, and highlights a critical weakness in systems previously thought to be more robust [VCRS20].

Building on this foundation, Schäfer et al. [SAH24] revisited and extended the work of Vidanage et al. Their contribution lies in a meticulous reproduction and replication of the original GMA, during which they identified a critical flaw: an undocumented pre-processing step in the provided codebase that inadvertently increased the effectiveness of the attack. While this step was originally intended to improve computational performance, it introduced errors into the proposed GMA. Schäfer et al. corrected this problem and further optimised the GMA, resulting in improved robustness and efficiency. Their improved implementation achieved higher re-identification rates compared to the original approach. This improvement not only validates the vulnerabilities highlighted by the GMA, but also highlights the potential for refining attack methodologies to expose even greater weaknesses in PPRL systems.

The work of Schäfer et al. is particularly relevant to this thesis, as their improved GMA implementation and accompanying codebase form the basis of the DEA proposed in this study. While the GMA is limited to re-identifying individuals present in both encrypted and plaintext datasets, the DEA seeks to extend the scope of re-identification beyond this intersection. Using ANNs trained on the re-identified individuals from the GMA, the DEA aims to predict and decode additional datasets, potentially leading to complete de-anonymisation of encrypted datasets.

To date, no existing research has proposed an approach comparable to the DEA. This thesis addresses this gap by developing and evaluating the DEA, thereby contributing to a broader understanding of PPRL vulnerabilities and highlighting the urgent need for more secure data linking techniques.

1.3. Contribution

The contribution of this thesis is divided into three main parts. First, a comprehensive analysis of PPRL systems is carried out, with particular emphasis on the three main encoding schemes: BF encoding, TSH encoding and TMH encoding. This analysis aims to highlight the

basic principles, strengths and weaknesses of each encryption scheme, setting the stage for the subsequent investigation of their susceptibility to DEA.

Next, the current state of the art GMA is analysed and its limitations are discussed in detail. Although GMAs have proven effective in re-identifying individuals within overlapping datasets, their applicability is limited to the intersection of plaintext and encrypted records. This inherent limitation highlights the need for more advanced attack strategies that can go beyond this.

The main focus of this thesis is the implementation and evaluation of the DEA, which attempts to outperform the capabilities of GMAs by decrypting a larger fraction of encrypted records. To achieve this, the thesis examines the conceptual foundations, theoretical underpinnings and technical requirements of the DEA. Building on the initial re-identifications made by the GMA, the DEA employs a supervised machine learning approach, specifically using ANNs trained on previously decoded data to predict and re-identify remaining encrypted records. This method significantly extends the scope of de-anonymisation in PPRL systems and provides a novel approach to current research.

The DEA is then evaluated against the three main PPRL encoding schemes. While the specific encoding scheme has minimal impact on the GMA, which is primarily based on solving a graph isomorphism problem, it plays a role in the DEA. This is due to the fact that the ANN has to be trained separately for each encoding scheme to account for the unique structural features and nuances of the encoding. However, the DEA is designed with adaptability in mind, ensuring that it can be effectively applied across different encoding schemes, thus increasing its generalisability and practical relevance.

Through this research, the thesis aims to answer critical questions about the robustness of PPRL systems. It investigates how effective supervised machine learning-based DEAs are at re-identifying the remaining entries that GMAs cannot decode. It also examines how different encoding schemes affect the performance and accuracy of the DEA, providing insight into which schemes are more susceptible to such attacks and why. By addressing these issues, the thesis contributes to a deeper understanding of the vulnerabilities inherent in PPRL systems and lays the groundwork for the development of more secure privacy preserving techniques.

1.4. Organization of this Thesis

This thesis is divided into four main sections: technical background, methodology, results, and conclusion.

First, an overview of PPRL systems is given, with particular emphasis on a thorough analysis of the most commonly used encoding techniques. Next, the existing GMA is introduced and explained in order to provide the basis for the study. In addition, an overview of ANNs is given to provide the necessary background knowledge.

Next, a detailed description of the attack model for the DEA is outlined, including how ANNs are used to enhance the attack. This is followed by an explanation of the actual implementation of the DEA, along with a discussion of the experiments conducted. The results of the DEA on different encryption schemes are then analysed and evaluated. Finally, the thesis concludes with a summary of the main contributions, a discussion of the broader implications, and suggestions for future research.

2. Background

This chapter provides an overview of the key concepts relevant to this thesis, including PPRL, various encoding techniques used in secure linkage and attacks. PPRL enables different organizations to link records belonging to the same individual across datasets while preserving privacy, making it a crucial tool. However, the security of such methods heavily depends on the encoding techniques employed to transform sensitive data into a more privacy-preserving format [VCRS20].

To understand the vulnerabilities of PPRL systems, we examine three key encoding techniques: BF, TMH, and TSH, each offering different trade-offs between efficiency, privacy, and robustness against attacks. While these methods aim to prevent direct access to plaintext identifiers, they remain susceptible to adversarial techniques designed to infer or reconstruct the original data [SAH24; VCRS20].

One such adversarial approach is the GMA, which leverages structural similarities between encoded and non-encoded datasets to re-identify individuals. Although GMAs are powerful in reconstructing intersections of datasets, they are limited to the intersection of the two datasets. To extend GMAs beyond known intersections, we explore ANN, which can learn complex patterns in encoded data and increase identification rates [SAH24; VCRS20].

By integrating these concepts, this chapter establishes the necessary foundation for understanding the DEA introduced later in this thesis, demonstrating how machine learning techniques can be employed to bypass existing privacy-preserving mechanisms.

2.1. Overview of PPRL

PPRL enables the linkage of records from different databases that refer to the same individual while preserving privacy. Traditional record linkage relies on unique identifiers, but these are often unavailable or inconsistent due to variations in formatting, spelling, or missing data across different databases. Linking records directly on plaintext data poses significant privacy risks. To mitigate these risks and prevent further threats to data security, regulations such as the European Union's General Data Protection Regulation have been established to govern the handling of PII [SAH24; VCRS20].

To enable record linkage using PII while preserving privacy, PPRL employs similarity-preserving encoding on quasi-identifiers, allowing linkage to be performed on encoded representations rather than raw data. This approach protects identities while still facilitating record matching based on encoded similarities and probabilistic approaches [SAH24; VCRS20].

The security of PPRL schemes depends on the encoding techniques used. Broadly, PPRL methods fall into two categories: perturbation based techniques and Secure Multi-Party Computation (SMC) based techniques. SMC-based techniques provide strong security guarantees and high accuracy but suffer from computational and communication overheads. Conversely, perturbation-based techniques balance linkage quality, scalability, and privacy protection, making them more practical for real-world applications [VCRS20].

A typical PPRL system involves three main parties working together to enable the linkage while maintaining privacy. The data owners, who are responsible for maintaining their respective databases, D and D', encode the quasi-identifiers by agreeing on an encoding scheme with the corresponding parameters before sharing them with the linkage unit. The linkage unit, a trusted entity, performs the actual record linkage using only the encoded representations, without access to the original identifiers. Once the linkage is completed, the linkage unit assigns unique pseudonyms to the successfully linked records and returns the pseudonymized dataset to the data owners. The data owners then replace the linkage data with these pseudonyms before sending the dataset to the data analysts. The analysts can merge the records based on these identifiers and proceed with further research and analysis, all while minimizing the risk of re-identification and protecting individual privacy [SAH24].

Each database record can therefore during the linkage process be represented as $r = (\lambda, \sigma)$, where λ denotes the linkage data which are encoded quasi-identifiers such as names and birthdates and σ refers to the remaining microdata like in a health care scenario patient information [SAH24].

Linkage in PPRL is performed probabilistically, meaning that two records, r and r', are considered linked if their similarity score on $sim(\lambda_r, \lambda_{r'})$ exceeds a predefined threshold. The choice of threshold plays a crucial role in balancing the quality of the linkage. Lower thresholds tolerate more variation and matches between records but increase the likelihood of false positives, while higher thresholds reduce false positives but may miss legitimate matches. Thus, selecting the optimal threshold is a trade-off that requires careful consideration of the specific goals and constraints of the linkage process [SAH24].

A robust PPRL scheme must satisfy several important criteria to ensure effective and secure linkage. First, a similarity function, $sim(\lambda_r, \lambda_{r'})$, must exist to determine if two records belong to the same entity based on a predefined threshold. Second, an encoding scheme must be applied to λ in such a way that the linkage unit cannot reconstruct the original data. Finally, a function $sim(enc(\lambda_r), enc(\lambda_{r'}))$ must be available that allows similarity computations on encoded data which preserved similarty properties. These requirements ensure both the privacy and the effectiveness of the record linkage process [SAH24].

One common approach for measuring similarity for two quasi identifiers is using n-grams. Here, string values are divided into overlapping substrings of length n using a sliding window approach [SAH24]. For example, for the string "hello" with $n = 2$, the n-grams are:

$$\{\text{he}, \text{el}, \text{ll}, \text{lo}\}$$

The similarity of two sets of n-grams can then be computed using metrics such as the Dice Coefficient [SAH24]

$$Dice(X, Y) = \frac{2 \times |X \cap Y|}{|X| + |Y|}$$

or using the Jaccard Similarity [SAH24]

$$Jaccard(X, Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

PPRL has been successfully applied in various domains, demonstrating its practical importance in securely linking records across institutions while preserving privacy. Notable applications include the Social Investment Data Resources (SIDR), the Lumos Initiative, the Swiss

National Cohort, and the Gemeinsamer Bundesausschuss (GBA). These implementations highlight the versatility of PPRL in diverse contexts, where privacy protection is essential while enabling effective data linkage for research and analysis [SAH24].

2.2. Key Encoding Techniques

In the context of PPRL, three primary encoding techniques have emerged BF, TMH and TSH. These encoding methods are essential for transforming sets of quasi-identifiers into fixed-length representations that preserve similarity information while maintaining privacy [SAH24; SBR09; VCRS20].

PPRL typically involves encoding sets of quasi-identifiers before linkage. A set in this context is generally a collection of n-grams, which are substrings of length n extracted from one or multiple attributes using a sliding window approach. Since input data varies in length, all encoding techniques in PPRL must take arbitrarily long inputs (sets of n-grams) and produce fixed-length encoded outputs. This ensures that similarity computations can be efficiently performed on encoded data by the linkage unit without accessing the raw identifiers [SAH24; VCRS20].

Before linkage, data owners must agree on the encoding scheme and share necessary cryptographic secrets to facilitate secure comparison. Among the available techniques, BFs are the most widely used approach for record linkage [SAH24].

2.2.1. BFs

BFs were originally developed for efficient membership testing in set structures without requiring direct access to the sets themselves [Blo70]. Due to their ability to efficiently compute set similarities in a privacy-preserving manner, they have been widely adopted in PPRL applications [SAH24; SBR09; VCRS20].

A BF $b \in \{0,1\}^l$ is a bit vector of length l. It uses $k \geq 1$ independent hash functions $H = \{h_1, h_2, ..., h_k\}$, where each function maps an arbitrary input to a position in the filter [SAH24; SBR09]:

$$h_i : \{0,1\}^* \rightarrow \{1, ..., l\}, \quad \forall i \in \{1, ..., k\} \tag{2.1}$$

Initially, the BF is set to all zeros. Each element $s \in S$ is hashed with every function h_i, and the corresponding bit positions in the filter are set to 1 [SAH24; SBR09]:

$$\forall s \in S, \forall h_i \in H, \quad b[h_i(s)] = 1 \tag{2.2}$$

Since BFs are binary vectors, the similarity between two BFs, b_1 and b_2, is computed based on the overlapping 1-bits based on their position. The Dice Coefficient is commonly used for this purpose [SAH24], providing a measure of similarity by comparing the overlap of 1-bits in the two binary vectors. This method allows for efficient computation of similarity between BFs, needs to be applied in the context of PPRL.

$$\text{Dice}(b_1, b_2) = \frac{2 \cdot |b_1 \cap b_2|}{|b_1| + |b_2|} \tag{2.3}$$

In PPRL, the set S consists of n-grams generated from the quasi-identifiers λ. This creates a deterministic relationship between the n-grams present and the set bits in the BF. However, due to the finite length of BFs, collisions occur where different n-grams map to the same

bit position. While this can cause incorrect linkages, it also enhances privacy by distorting frequency distributions [SAH24; VCRS20].

Three primary approaches exist for applying BFs to sensitive data. The first approach, Attribute-Level BFs, encodes each attribute, such as first name or last name, into a separate BF, enabling multiple similarity computations. However, Attribute-Level BFs are more vulnerable to frequency-based privacy attacks. The second approach, Cryptographic Long-Term Key Encoding, merges multiple attributes into a single BF, reducing vulnerability to frequency attacks but remaining susceptible to pattern-mining-based attacks. Finally, Record-Level BFs employ a weighted bit sampling technique to minimize frequency information, enhancing privacy protection while maintaining high linkage quality. Each of these approaches balances privacy concerns with the need for accurate and effective record linkage [VCRS20].

Several privacy-enhancing methods have been proposed to mitigate frequency attacks in BFs. These techniques introduce a trade-off between privacy and linkage quality. One such method is balancing, which ensures an equal number of 1-bits across BFs, thereby reducing the likelihood of frequency-based attacks. Another approach is salting, which randomizes bit positions to prevent direct inference from the BFs. Additionally, XOR folding is used to reduce the BF length while maintaining the bit-wise dependencies necessary for effective linkage. These methods aim to strengthen privacy while retaining the accuracy of the linkage process [SAH24; VCRS20].

A major improvement was introduced by Armknecht et al. [AHS23], who proposed a diffusion layer for BF encodings. This method generates Encoded Linkage Data (ELD), where each bit is computed as the XOR sum of multiple BF bits. The indices for XOR computations are randomly chosen and secretly shared among data owners [AHS23].

By applying diffusion, the deterministic relationship between 1-bits in the ELD and the original n-grams is broken, improving privacy while still enabling approximate matching [AHS23].

BFs remain a core technique in PPRL due to their efficiency and scalability. However, their vulnerability to frequency attacks has led to improvements such as Record-Level BFs and diffusion layers, which enhance privacy at the cost of increased computational complexity [AHS23; SAH24; VCRS20].

2.2.2. TMH

TMH is a variation of MinHash initially introduced for efficient estimation of set similarities and later adapted for privacy-preserving probabilistic record linkage. MinHash itself was first proposed in the context of document resemblance and containment estimation. TMH extends MinHash by employing tabulation-based hashing, which enhances its security compared to BFs [Bro97; VCRS20].

MinHash aims to approximate the Jaccard similarity between two sets, S and S'. The fundamental idea is to represent both sets as sequences of randomly ordered elements and apply multiple rounds of random permutations, π and π', to shuffle them. After each round, the first elements of both sequences are compared. The larger the intersection between S and S', the higher the probability that the first elements will match. The final Jaccard similarity estimate is computed based on the number of hash collisions achieved during permutation [Bro97; SAH24; VCRS20].

Instead of explicitly computing these permutations, MinHash simulates them by applying a suitable hash function to the elements of a set and selecting the smallest hash value as the representative signature. This is equivalent to sorting set elements by their hash values and

returning the first element [SAH24].

Tabulation-based hashing is a technique used in TMH that provides efficient and high-quality hash functions by leveraging precomputed lookup tables. This method operates as follows [VCRS20]:

The process begins with the **initialization** of l sets of lookup tables, each containing c tables. Each table holds randomly generated bit strings for keys of length k, with a key space of 2^k. During the **hashing process**, each element in S is hashed using a one-way hash function, producing a fixed-length binary value. This binary value is then split into c sub-keys, each of length k. Each sub-key is used as an index to retrieve a random bit string from the corresponding lookup table, and the retrieved c bit strings are XORed together to produce a single output bit string. In the **MinHash signature generation** phase, this process is repeated for each of the l lookup table sets, and the minimum value among all generated bit strings is selected as the MinHash signature [SAH24; VCRS20].

To further enhance privacy, TMH employs a 1-bit hashing mechanism, where only the least significant bit of each MinHash signature is retained. These l bits are then concatenated to form the final bit array used as an encoded representation [SAH24].

The main advantage of TMH over BFs is its improved resistance to frequency-based attacks due to the complexity introduced by tabulation-based hashing. However, this security enhancement comes at a cost. It leads to higher computational overhead, as the need to generate and access multiple lookup tables increases processing time. Additionally, there is increased memory consumption because storing large precomputed tables requires additional space. These trade-offs must be considered when choosing between TMH and other privacy-preserving techniques [SAH24; VCRS20]. Despite these trade-offs, TMH remains an attractive alternative for PPRL due to its robustness against adversarial attacks.

Similar to BFs, TMH encodes each record as a bit vector of length l. Given two TMH-encoded bit vectors, their similarity can be estimated using a modified Jaccard coefficient, adapted to account for artificial bit collisions caused by truncation to the least significant bit. The Jaccard coefficient can also be converted into the Dice coefficient for improved comparability with BF-based methods [SAH24; VCRS20].

Overall, TMH provides a more secure encoding alternative to BFs in PPRL, though at the expense of increased computational and memory requirements [SAH24; VCRS20].

2.2.3. TSH

TSH is the most recent encoding scheme proposed for PPRL , introduced in 2020 [VCRS20]. TSH was designed to address both the privacy vulnerabilities of BFs and the computational complexity of TMH while maintaining accuracy in similarity calculations. Similar to other encoding techniques, TSH requires the input to be split into a set of n-grams S prior to encoding [SAH24].

As a result of TSH encoding, each record from a sensitive database is represented by a set of integers, which can be directly used to compute Jaccard similarity. TSH employs two distinct hashing steps. In the first hashing step, the input set is converted into a bit matrix representation. In the second hashing step, the bit matrix columns are mapped into integers, enabling efficient comparison. This two-step process allows TSH to represent sensitive data in a way that facilitates effective similarity computation while preserving privacy [SAH24]. These steps provide accurate Jaccard similarity calculations while improving privacy protection compared to traditional BF-based encodings [VCRS20].

In the first step of the TSH process, elements of the n-gram set S are hashed into k independent BFs b_i of length l, meaning each hash function results in a corresponding bit vector. This generates a $k \times l$ matrix, where each row corresponds to a BF created using a unique hash function, and each column represents the bitwise state across all BFs for a given position. In the second step, after constructing the bit matrix, TSH computes column-wise hashes to convert the bit vectors into integer representations . Each column vector is treated as an input for a hash function, and all-zero columns are skipped to prevent distortion in similarity calculations since they do not encode any n-grams. To enhance security and avoid hash collisions between columns with identical bit patterns, a salt value and the column index are concatenated before hashing [SAH24; VCRS20].

The final integer representation for each column i for $1 \leq i \leq l$. is computed as [SAH24]:

$$H(salt, i, b_{1i}, b_{2i}, ..., b_{ki}) \tag{2.4}$$

The output of the second hashing step is a set of integers, allowing similarity computations using the Dice coefficient rather than directly computing bitwise similarity. Since the encoded data consists of sets, Jaccard similarity can also be computed similarly to MinHash-based encodings [SAH24].

To improve efficiency, TSH can be implemented using a Pseudo-Random Number Generator (PRNG) instead of cryptographic hash functions. The PRNG is seeded with the value to be hashed before generating random numbers, ensuring that the sequence of generated values depends deterministically on the input [SAH24].

By combining efficient bit vector representations with integer-based similarity computations, TSH offers a balance between privacy, security, and computational efficiency, making it a promising alternative to existing PPRL encoding schemes [SAH24; VCRS20].

2.3. GMAs

GMAs were first introduced by Vidanage et al. [VCRS20] and represent the most significant threat to PPRL due to their universal applicability. Unlike traditional cryptanalytic attacks, GMAs exploit the fundamental properties of non-interactive PPRL to compromise the security of all schemes relying on similarity-preserving encoding [SAH24].

Non-interactive PPRL refers to linkage schemes where data owners independently encode their data and share it with a linkage unit. The linkage unit then performs record matching solely based on the encoded data, without requiring further interaction with the data owners during the linkage process. This approach minimizes communication overhead and computational complexity, as no iterative exchanges between parties are necessary. In contrast, interactive PPRL methods involve multiple rounds of communication between data owners and the linkage unit to refine matching results or improve accuracy [**kum2014privacy**].

In PPRL, encoded records are linked based on similarity computations. Since these similarities serve as identifiers, an attacker with access to both encoded and plaintext data can leverage them to re-identify individuals. The latest version of the GMA developed by Schaefer et al. overcomes the limitations of the original attack by Vidanage et al. and enhances success rate and robustness, even under limited knowledge scenarios [SAH24].

In the context of a PPRL system, the attacker is modeled as the linkage unit and is assumed to have minimal prior knowledge. The attacker does not know any encoding secrets, seeds, or salts used in the system to protect the data. The only information available to the attacker is

that which is inevitably known to the linkage unit during the linkage process. This assumption ensures that the attacker can only exploit data accessible through normal system operations, adhering to Kerckhoffs's principle [SAH24].

Since the attack does not depend on specific encoding parameters or attribute frequency distributions, it is universally applicable as long as pairwise similarities of encoded data are available [SAH24].

The first step of the attack involves constructing similarity graphs for both the encoded dataset (D_{enc}) and the plaintext dataset (D_{plain}). In these graphs, each node represents an individual record, while edges between nodes are assigned weights based on pairwise similarity computations. To ensure computational efficiency and focus only on meaningful connections, edges with similarity scores below a predefined threshold are omitted, reducing noise and improving the accuracy of the attack [SAH24].

Since certain encoding properties, such as BF length (l), are inevitably known to the linkage unit, D_{plain} can be transformed analogously to D_{enc} for effective comparison. Importantly, this step does not require knowledge of shared secrets, as the primary objective is to replicate the effect of encoding on similarity [SAH24].

To quantify the structural similarity between nodes in G_{plain} and G_{enc}, node embeddings are computed to transform the graph structure into a numerical representation. This process begins with graph embedding using the Node2Vec algorithm, which applies a Word2Vec-like approach to learn vector representations of nodes. During this process, nodes undergo multiple random walks, where each walk simulates a sequence of transitions between connected nodes. These sequences are then treated as sentences, allowing the model to learn embeddings that capture the local and global structure of the graph. The behavior of these random walks is controlled by two hyperparameters: p, which determines the likelihood of returning to a previously visited node, and q, which influences the tendency to explore new regions of the graph. The result is an embedding matrix where each row represents a node as a vector in Euclidean space [SAH24].

Once embeddings are generated, they must be aligned to allow meaningful comparison between the two graphs. Due to the randomness inherent in embedding generation, direct comparison is not possible. Instead, an iterative approach is used to solve two subproblems: first, an optimal linear transformation is determined using Procrustes Analysis to align the embeddings, and second, node correspondences are established via the Sinkhorn Algorithm, which minimizes the Wasserstein distance between the distributions of embeddings in both graphs. To achieve an effective alignment, an unsupervised stochastic optimization scheme alternates between these two steps over n epochs, gradually refining the transformation and correspondences until convergence [SAH24].

Once the embeddings from the plaintext and encoded datasets are aligned, the reidentification process can begin. Each embedding in the transformed plaintext space is compared to its counterparts in the encoded space, with similarity measured using cosine similarity. This metric quantifies how closely two embeddings align in the high-dimensional space, enabling the attacker to identify records in the encoded dataset that most closely resemble those in the plaintext dataset [SAH24].

The final step involves constructing a bipartite graph, where nodes from the plaintext and encoded datasets are linked based on their similarity scores. To determine the optimal mapping, the Jonker-Volgenant algorithm is applied, ensuring that each node in the smaller dataset is uniquely matched to a corresponding node in the larger dataset. This algorithm maximizes the total similarity across all matched pairs, effectively revealing the identities of individuals

within the encoded dataset [SAH24].

The novel GMA approach by Schaefer et al. achieves near-perfect re-identification rates when dataset overlap is 100%. Even for low-overlap scenarios (e.g., 5%), success rates reach 99.9% for TSH [SAH24]. The only encoding scheme resistant to GMAs is BFs with diffusion layers, which disrupts similarity preservation for sufficiently high diffusion values [SAH24].

2.4. ANNs

3. Methodology

The **Dataset Extension Attack (DEA)** is a novel attack method that extends the capabilities of **Graph Matching Attacks (GMA)** by going beyond the intersection of datasets and attempting to re-identify previously unmapped individuals. This chapter details the methodology behind the DEA, including the necessary modifications to the GMA, the design and implementation of the DEA itself, and the role of neural networks in enabling probabilistic reconstruction of **Personally Identifiable Information (PII)** from encoded representations.

The DEA builds upon the GMA by utilizing its re-identification results as a foundation for further inference. Since the GMA only establishes correspondences between records that exist in both the attacker's auxiliary dataset and the encoded target dataset, it leaves a significant portion of records unmapped. The goal of the DEA is to extend this re-identification process by leveraging a machine learning-based approach to infer missing **2-grams**—the fundamental building blocks of encoded names in the Bloom filter-based privacy-preserving record linkage (PPRL) scheme used in this research.

To achieve this, the DEA follows a structured pipeline consisting of several key steps. First, the results of the GMA are extracted in a predefined format to serve as training data for the neural network. These results include re-identified individuals, their corresponding encoded representations, and the plaintext names that were successfully mapped. Ensuring a structured and consistent format is crucial, as it allows for seamless integration into the subsequent processing stages.

Once the data is extracted, it is transformed into a format suitable for neural network training. This involves setting up specialized datasets that convert encoded representations and their corresponding labels (plaintext **2-grams**) into tensor-based formats that can be efficiently processed by deep learning models. The dataset is then split into **training, validation, and test sets**, and respective **data loaders** are created to facilitate efficient batch processing during training.

With the data pipeline in place, a neural network is designed to perform **multi-label classification**, predicting the likelihood of individual **2-grams** occurring in an encoded record. The architecture consists of an **input layer** that receives the encoded representation, a series of **hidden layers** that learn complex patterns within the data, and an **output layer** that produces probability scores for all possible **2-grams**. To optimize performance, an appropriate **loss function** and **optimizer** are selected, ensuring stable convergence during training. The model is trained using the training set while the validation set is used to monitor performance and prevent overfitting.

Once the neural network is trained, it is applied to the set of **not-reidentified individuals**, i.e., records that remained unmapped after the GMA. The model outputs a probability distribution for each possible **2-gram**, indicating the likelihood of its presence in the corresponding plaintext name. To refine these predictions, a **thresholding mechanism** is applied, filtering out low-confidence predictions and retaining only the most relevant **2-grams**. Finally, the predicted **2-grams** are aggregated and reconstructed into potential **PII**, forming the final step of the **DEA** process.

This methodological approach represents a significant step forward in attacking **privacy-preserving record linkage (PPRL)** systems. By leveraging deep learning techniques, the DEA enables an attacker to infer sensitive personal information beyond the scope of traditional GMA approaches. The following sections provide an in-depth discussion of each component, including the design choices, implementation details, and challenges encountered during development.

[Let me know when you're ready for the next section or if you want to refine anything!]

The **Graph Matching Attack (GMA)** serves as the foundation for the **Dataset Extension Attack (DEA)** by providing the initial set of re-identified individuals along with their corresponding encodings. The effectiveness of the DEA is directly dependent on the performance of the GMA. A higher re-identification rate in the GMA results in a larger training dataset for the DEA, which improves its ability to infer missing **2-grams** and reconstruct additional identities. Conversely, if the GMA has a low re-identification rate, the DEA is constrained by the limited amount of labeled training data, reducing its overall effectiveness in identifying individuals beyond the intersection of datasets.

To integrate the GMA as a preprocessing step for the DEA, modifications were made to the original implementation by **Schaefer et al.** While the core algorithm remains unchanged, adjustments were necessary to ensure that the GMA outputs results in a structured format that facilitates the training of the neural network used in the DEA. Originally, the GMA only provided a simple mapping between IDs of re-identified individuals. However, for the DEA to be able to learn meaningful patterns, it requires access to both the **plaintext PII and their corresponding encodings**. Therefore, modifications were implemented to ensure that the GMA outputs data in the following format:

- For **re-identified individuals**: '<PII> <encoding> <uid>' - For **not-reidentified individuals**: '<encoding> <uid>'

Here, the **uid** is included only for research and testing purposes. It allows the developer to manually track individuals across different processing stages. However, in a real-world attack scenario, these **uids** are neither available nor necessary. They are entirely excluded from any DEA training or inference steps, ensuring that the attack methodology remains realistic and applicable in practical settings.

In addition to formatting adjustments, some components of the **GMA** were removed to streamline the process and reduce unnecessary complexity. Specifically, encoding schemes such as **Two-Step Hashing (TSH)** and **Tabulation MinHash (TMH)** were excluded, as the focus of the **DEA** remains solely on **Bloom Filter (BF) encodings**. These optimizations resulted in a leaner and more efficient attack pipeline, reducing computational overhead while retaining all essential functionality.

With these modifications in place, the **starting point for the DEA** is clearly defined. The attack begins with two structured datasets:

1. **Re-identified individuals**, containing both their **plaintext PII and corresponding encodings**, formatted as described above. 2. **Not-reidentified individuals**, for whom only the **encodings** are available, serving as the primary targets for inference using the DEA's neural network model.

By leveraging this structured output, the DEA is able to train a machine learning model capable of probabilistically reconstructing missing **2-grams** from the encoded records of not-reidentified individuals. The following sections will detail the implementation of this approach, including dataset preparation, model architecture, and evaluation strategies.

3.1. Design and Implementation of the DEA

The Dataset Extension Attack (DEA) aims to reconstruct plaintext Personally Identifiable Information (PII) from encoded records by leveraging machine learning techniques. This section provides a detailed account of the problem definition, data representation, neural network architecture, and training methodology. A key challenge in the implementation of the DEA is the diversity of encoding schemes used to protect sensitive data. Since different encoding methods transform plaintext into distinct numerical representations, the design of the DEA must account for these variations by tailoring the dataset structure and neural network architecture to each encoding scheme.

To address this challenge, the DEA follows a modular approach, where the general attack methodology remains consistent, but specific implementations are adapted for each encoding scheme. While the input representation and network architecture differ based on the encoding method, the output format remains uniform across all models. The attack is formulated as a multi-label classification problem, where the neural network predicts the probability of individual 2-grams being present in the original PII. For each encoding scheme, a dedicated dataset structure is constructed to transform encoded records into an appropriate numerical format suitable for neural network training.

3.1.1. Problem Definition

The primary challenge that the **Dataset Extension Attack (DEA)** seeks to address is the **limited scope of re-identifications** achieved by the **Graph Matching Attack (GMA)**. While the GMA is effective in linking records by exploiting the structural relationships within encoded datasets, its success is inherently constrained to individuals who are present in both datasets and can be matched based on graph similarity. However, in many real-world scenarios, there exists **additional re-identification potential** beyond these direct matches.

One possible approach to extending re-identifications is to incorporate **publicly available data** and rerun the GMA in an iterative manner, gradually refining the matching process. However, this approach is inherently dependent on external data sources and may not be viable in cases where such additional information is scarce. Instead, the **DEA introduces a novel method that reconstructs deterministic relationships between encoded representations and their corresponding plaintext information**, leveraging the fact that **all encoding schemes used in Privacy-Preserving Record Linkage (PPRL) rely on hash functions**.

Hash functions provide a **fixed-length output** for an **arbitrary-length input**, and crucially, they are **deterministic**, meaning that the same input will always produce the same hash. The **DEA exploits this property** by training neural networks to **learn statistical relationships between the encoded values and the original 2-grams** of **Personally Identifiable Information (PII)**. The goal is to recover the most probable plaintext representation given an encoded input, effectively treating the attack as a **probabilistic frequency-based inference problem**.

However, several factors make this task inherently difficult. The first challenge is that **the exact number and type of hash functions used in the encoding process are unknown**. This means that the model must learn patterns without explicit knowledge of the hashing mechanisms applied. Fortunately, this limitation is mitigated by the fact that the **DEA does not require a one-to-one mapping** between hash outputs and plaintext but instead relies on statistical inference across multiple samples.

A more fundamental obstacle arises from the **collision property of hash functions**. Since hash functions **map an infinite input space to a finite output space**, different inputs may produce **identical hashes**, making it difficult to perfectly recover the original plaintext values. These collisions introduce inherent **uncertainty** into the re-identification process, preventing the DEA from achieving **100% accuracy**. As a result, the DEA's predictions are **probabilistic rather than deterministic**, meaning that the attack **can estimate the likelihood of a given 2-gram being present in the original PII but cannot guarantee absolute correctness**.

The primary reason the **GMA alone fails to achieve this kind of re-identification** is that it **relies solely on the structural properties of the dataset**, without attempting to infer direct relationships between encoded values and their plaintext counterparts. By contrast, the DEA **extends the capabilities of the GMA by reconstructing individual plaintext components from encoded representations, thereby increasing the overall re-identification potential**. This novel approach significantly enhances the attack's effectiveness, allowing for the possibility of re-identifying individuals who were previously considered unmatchable using traditional graph-based techniques.

3.2. Data Representation

Data Representation

For a neural network to function effectively and achieve successful results, it is essential to **preprocess the data into a format that is consumable by deep learning models**. This preprocessing applies to both **input data**, which consists of encoded representations of personally identifiable information (PII), and **output data**, which represents the predicted 2-grams. Since artificial neural networks (ANNs) in PyTorch operate on **tensor representations**, the transformation of encoded records into tensors is a crucial step. This transformation ensures that both re-identified and not-reidentified individuals are structured in a way that facilitates efficient training and inference. To achieve this, **PyTorch datasets** are created to handle the conversion of encoded inputs into tensors while also encoding the expected output in a suitable multi-label classification format.

Structure of Encoded Records and Their Representations

The data structure follows the encoding schemes previously described, with different encoding methods leading to different preprocessing techniques. Each encoding type requires a tailored transformation into tensors, ensuring compatibility with the neural network architecture while preserving as much information as possible.

Bloom Filter (BF) Encoding: Bloom filters are fixed-length binary strings, with their length determined by Alice's chosen parameters. These bitstrings are converted into PyTorch tensors of the same length. The transformation is straightforward—**each bit in the Bloom filter is mapped directly to the tensor**, with the positions of **1-bits preserved**, ensuring that the structure of the encoding remains unchanged. The resulting tensor has the same length as the Bloom filter, with **1s at positions where bits were set in the original filter and 0s elsewhere**.

Tabulation MinHash (TMH) Encoding: Tabulation MinHash, like Bloom Filters, also produces **fixed-length binary bitstrings**, where the length depends on the parameters chosen by Alice. The transformation process follows the same principles as for Bloom Filters: **each TMH bitstring is converted into a PyTorch tensor of equal length, preserving the po-

sitions of the 1-bits**. This ensures that the neural network receives the TMH encoding as a structured binary representation.

Two-Step Hashing (TSH) Encoding: The preprocessing of **Two-Step Hashing (TSH) encodings** is more complex due to its **variable-length representation**. Unlike BF and TMH, TSH **does not produce fixed-length binary bitstrings** but rather a **set or list of integers**. However, neural networks require **fixed-length input vectors**, meaning that an appropriate transformation must be applied. Aggregation techniques (such as computing averages) would lead to information loss, which is undesirable in this limited-knowledge setting. Instead, two different approaches are considered for transforming TSH into a tensor-compatible format:

1. **Padding-Based Approach:** The first method involves determining the **largest set size** among all TSH-encoded records. This maximum length is then used to define a **fixed-size tensor representation** for all samples. Each set of TSH integers is mapped into a tensor where **the original values are placed at the beginning**, and **any remaining positions are padded with zeros**. For example, if the largest set size is **5**, and two different TSH sets are ‘22, 3, 4‘ and ‘11, 8‘, they are transformed into ‘[22, 3, 4, 0, 0]‘ and ‘[11, 8, 0, 0, 0]‘, respectively.

2. **Bitstring Mapping Approach:** The second method transforms the TSH encoding into a **bitstring format similar to BF and TMH**. In this case, the **maximum integer value** encountered in any TSH encoding (e.g., **12345** if that is the highest number present) determines the tensor length. Each TSH integer is then mapped to a corresponding index, setting that position in the tensor to **1**. If a specific value appears multiple times within a TSH set, the corresponding index in the tensor is **incremented instead of simply being set to 1**. This ensures that frequency information is retained, reducing the risk of losing critical details due to hash collisions. For instance, if ‘22‘ appears twice in a TSH encoding, the tensor at index ‘22‘ will have the value **2** rather than **1**.

Regardless of the encoding scheme used, **the output of the neural network remains the same**. The goal is to **map the encoding input to the probability distribution of 2-grams**, which means that the **output layer always predicts the likelihood of each possible 2-gram being present in the original plaintext**.

Re-Identified Individuals as Labeled Training Data

To enable supervised learning, **re-identified individuals** are used as **labeled training data**. Since their **PII is known along with their corresponding encoded representation**, it is possible to create a **training dataset** where the input consists of transformed encodings (BF, TMH, or TSH) and the output consists of **the correct 2-grams derived from the original PII**.

To facilitate this process, a **predefined dictionary of all possible 2-grams** is constructed. This dictionary includes: - **Alphabetical 2-grams** (‘aa‘ to ‘zz‘) - **Numerical 2-grams** (‘00‘ to ‘99‘) - **Mixed alphanumeric 2-grams** (‘a0‘ to ‘z9‘)

Since the datasets used in this research primarily contain **first names, last names, and birthdates**, these character sets cover the majority of cases. Each **possible 2-gram is mapped to a specific index** in the output tensor, ensuring a consistent representation across training samples. For example, if index ‘1‘ corresponds to the 2-gram **"ab"**, and the neural network predicts a **60

By structuring the data in this way, the neural network is trained to **map the encoding to its corresponding 2-gram representation**, ultimately enabling the DEA to probabilistically reconstruct the original PII from encoded data.

3.2.1. ANN Architecture for DEA

Attempting to reconstruct plaintext information from encoded representations based on hash functions presents a significant challenge due to the nature of cryptographic hashing. Since hash functions are designed to be one-way functions, reversing the transformation to recover the original input is theoretically infeasible. However, while exact reconstruction is not possible, a probabilistic approach can still be applied to infer likely plaintext components based on patterns in the encoded data.

Neural networks provide a powerful framework for learning complex mappings between input encodings and output predictions, making them well-suited for this task. The function of the neural network in the DEA is to predict 2-grams by learning from re-identified individuals—individuals whose plaintext information is known along with their corresponding encoding. Through this learning process, the model captures frequency patterns that emerge due to the deterministic properties of hash functions. In essence, the neural network learns which 2-grams are mapped to specific positions within the encoding schemes and leverages these patterns to estimate the probable presence of certain 2-grams in encoded records.

Although hash functions introduce collisions, meaning that different inputs can map to the same encoded value, the neural network can still extract meaningful probabilistic insights by recognizing the most common mapping relationships across training data. This allows the DEA to provide a ranked list of likely 2-grams, forming the basis for the reconstruction of personally identifiable information (PII) from partially anonymized data.

3.2.1.1. Neural Network Architecture for Each Encoding Scheme

The architecture of the neural network varies depending on the encoding scheme used, as each encoding method results in a different type of input representation. However, the output layer remains consistent, as the goal across all models is to predict the probability distribution over the set of possible 2-grams.

For the Bloom Filter (BF) model, the input layer corresponds to the length of the Bloom Filter bitstring, which is determined by Alice's parameter choices. The network consists of hidden layers that process the encoded information and extract patterns relevant to 2-gram mapping. The output layer has a fixed size equal to the number of entries in the 2-gram dictionary, with each neuron representing the probability of a specific 2-gram being present in the original plaintext.

Similarly, the Tabulation MinHash (TMH) model follows the same structure, with an input layer corresponding to the length of the TMH bitstring, hidden layers that adaptively learn feature representations, and an output layer of the same fixed size as the 2-gram dictionary.

For the Two-Step Hashing (TSH) model, the input layer configuration depends on the transformation approach selected. One approach, padding, sets the input layer size to the largest set size among all TSH-encoded records, ensuring a uniform input size. The other approach, bitstring mapping, determines the input size based on the largest integer value encountered across all TSH encodings, constructing an input representation similar to Bloom Filters but with an extended numerical range. Regardless of the transformation approach, the output layer remains consistent in predicting the likelihood of each 2-gram.

3.2.1.2. Choice of Activation Functions, Loss Function, and Optimizer

To train the neural networks effectively, specific activation functions, loss functions, and optimizers are chosen based on the nature of the task. Since the task is multi-label classification—where each sample may contain multiple valid 2-grams—the Binary Cross Entropy with Logits Loss (extttBCEWithLogitsLoss) is selected. This loss function is well-suited for problems where the model needs to predict independent probabilities for multiple classes rather than a single categorical output.

For optimization, the Adam (Adaptive Moment Estimation) optimizer is chosen due to its ability to handle sparse gradients efficiently and its robustness in optimizing deep neural networks. Adam dynamically adjusts the learning rate for each parameter, leading to faster convergence compared to standard stochastic gradient descent (SGD).

The selection of hidden layer architectures and activation functions remains an open area for experimentation and hyperparameter tuning to optimize performance across different encoding schemes.

3.2.2. Training the Model

To effectively train and evaluate the neural network models, the dataset is divided into three distinct subsets: a training set, a validation set, and a test set. The training set consists of 80% of the labeled dataset, while the remaining 20% is designated as the validation set. The test set comprises the not-reidentified individuals, serving as the primary evaluation set for the trained model.

Dataloaders are created for each of these subsets to facilitate efficient mini-batch processing. Different batch sizes are employed depending on the dataset subset to optimize computational performance and convergence behavior. The training and validation dataloaders enable efficient iteration over the respective data splits, ensuring that the neural network is exposed to all available samples during training and validation.

3.2.2.1. Loss Computation and Performance Metrics

The performance of the neural network models is assessed using multiple evaluation criteria. The primary metrics include training loss, validation loss, and various classification performance indicators. The loss function employed during training is Binary Cross Entropy with Logits Loss (BCEWithLogitsLoss), which is well-suited for multi-label classification tasks.

During training, the loss is computed for each mini-batch, and the cumulative loss across the entire training dataset is used to track the optimization progress. Similarly, validation loss is computed over the validation set at the end of each epoch to monitor generalization performance. Additional evaluation metrics such as accuracy, precision, and recall may be computed to assess the quality of the predictions. Furthermore, a re-identification rate is determined by applying a threshold-based approach using similarity metrics, enabling the assessment of the attack's effectiveness in re-identifying previously non-mapped individuals.

3.2.2.2. Training Process and Hyperparameter Tuning

The training process consists of multiple epochs, where each epoch involves iterating through the entire training dataset using the data loader. For each mini-batch, the model performs

a forward pass, computes the loss, and applies backpropagation to update the network's parameters using the Adam optimizer. After processing all training batches in an epoch, the model's performance is evaluated on the validation set to compute validation loss and monitor potential overfitting.

Hyperparameter tuning is an essential step to optimize the model's performance. Various hyperparameters, such as learning rate, batch size, and the number of hidden layers, are systematically adjusted and evaluated based on validation performance. Techniques such as grid search or random search may be employed to identify the optimal configuration. Regularization methods, including dropout and weight decay, may also be incorporated to prevent overfitting and improve generalization capabilities.

4. Results

4.1. Evaluation Metrics

4.2. Experiments

4.3. Analysis

4.4. Discussion

5. Conclusion

5.1. Summary

5.2. Future Work

Bibliography

[AHS23] Frederik Armknecht, Youzhe Heng, and Rainer Schnell. "Strengthening privacy-preserving record linkage using diffusion." In: *Proceedings on Privacy Enhancing Technologies* (2023).

[Blo70] Burton H Bloom. "Space/time trade-offs in hash coding with allowable errors." In: *Communications of the ACM* 13.7 (1970), pp. 422–426.

[Bro97] Andrei Z Broder. "On the resemblance and containment of documents." In: *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*. IEEE. 1997, pp. 21–29.

[FS69] Ivan P Fellegi and Alan B Sunter. "A theory for record linkage." In: *Journal of the American Statistical Association* 64.328 (1969), pp. 1183–1210.

[HSW07] Thomas N Herzog, Fritz J Scheuren, and William E Winkler. *Data quality and record linkage techniques*. Vol. 1. Springer, 2007.

[IH18] Jim Isaak and Mina J Hanna. "User data privacy: Facebook, Cambridge Analytica, and privacy protection." In: *Computer* 51.8 (2018), pp. 56–59.

[KM24] Jennifer King and Caroline Meinhardt. *Rethinking Privacy in the AI Era: Policy Provocations for a Data-Centric World*. White Paper, Stanford University Institute for Human-Centered Artificial Intelligence (HAI). 2024. URL: http://www.darkpatternstipline.org.

[MK19] Karl Manheim and Lyric Kaplan. "Artificial intelligence: Risks to privacy and democracy." In: *Yale JL & Tech.* 21 (2019), p. 106.

[PSZ+24] Aditi Pathak, Laina Serrer, Daniela Zapata, Raymond King, Lisa B Mirel, Thomas Sukalac, Arunkumar Srinivasan, Patrick Baier, Meera Bhalla, Corinne David-Ferdon, et al. "Privacy preserving record linkage for public health action: opportunities and challenges." In: *Journal of the American Medical Informatics Association* 31.11 (2024), pp. 2605–2612.

[RCS20] Thilina Ranbaduge, Peter Christen, and Rainer Schnell. "Secure and accurate two-step hash encoding for privacy-preserving record linkage." In: *Advances in Knowledge Discovery and Data Mining: 24th Pacific-Asia Conference, PAKDD 2020, Singapore, May 11–14, 2020, Proceedings, Part II 24*. Springer. 2020, pp. 139–151.

[SAH24] Jochen Schäfer, Frederik Armknecht, and Youzhe Heng. "R+R: Revisiting Graph Matching Attacks on Privacy-Preserving Record Linkage." In: *Proceedings of [Conference Name, if available]*. Available at: https://github.com/SchaeferJ/graphMatching. University of Mannheim. 2024.

[SBR09] Rainer Schnell, Tobias Bachteler, and Jörg Reiher. "Privacy-preserving record linkage using Bloom filters." In: *BMC medical informatics and decision making* 9 (2009), pp. 1–11.

[Smi16] Tanshanika T Smith. *Examining data privacy breaches in healthcare*. Walden University, 2016.

[Smi17] Duncan Smith. "Secure pseudonymisation for privacy-preserving probabilistic record linkage." In: *Journal of Information Security and Applications* 34 (2017), pp. 271–279.

[VCRS20] Anushka Vidanage, Peter Christen, Thilina Ranbaduge, and Rainer Schnell. "A graph matching attack on privacy-preserving record linkage." In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2020, pp. 1485–1494.

[VSCR17] Dinusha Vatsalan, Ziad Sehili, Peter Christen, and Erhard Rahm. "Privacy-preserving record linkage for big data: Current approaches and research challenges." In: *Handbook of big data technologies* (2017), pp. 851–895.

A. Auxiliary Information

Eidesstattliche Erklärung

Hiermit versichere ich, dass diese Abschlussarbeit von mir persönlich verfasst ist und dass ich keinerlei fremde Hilfe in Anspruch genommen habe. Ebenso versichere ich, dass diese Arbeit oder Teile daraus weder von mir selbst noch von anderen als Leistungsnachweise andernorts eingereicht wurden. Wörtliche oder sinngemäße Übernahmen aus anderen Schriften und Veröffentlichungen in gedruckter oder elektronischer Form sind gekennzeichnet. Sämtliche Sekundärliteratur und sonstige Quellen sind nachgewiesen und in der Bibliographie aufgeführt. Das Gleiche gilt für graphische Darstellungen und Bilder sowie für alle Internet-Quellen.
Ich bin ferner damit einverstanden, dass meine Arbeit zum Zwecke eines Plagiatsabgleichs in elektronischer Form anonymisiert versendet und gespeichert werden kann.

Datum	Marcel Mildenberger